

DECEMBER 13, 2016

DRUG WATCH
PROJECT REPORT
CMPE 272 – ENTERPRISE SOFTWARE PLATFORMS

ARPITA DIXIT

CMPE 272

GitHub ID: arpitadixit
SJSU ID: 010953267

ADITI SHETTY

CMPE 272

GitHub ID: shettyaditi
SJSU ID: 011443016

SURAJ KHURANA

CMPE 272

GitHub ID: khurana3773
SJSU ID: 011464427

NRUPESH PATEL

CMPE 272

GitHub ID: Nrupesh29
SJSU ID: 011425271

GitHub Repository: <https://github.com/SJSU272Lab/Fall16-Team11>

INTRODUCTION

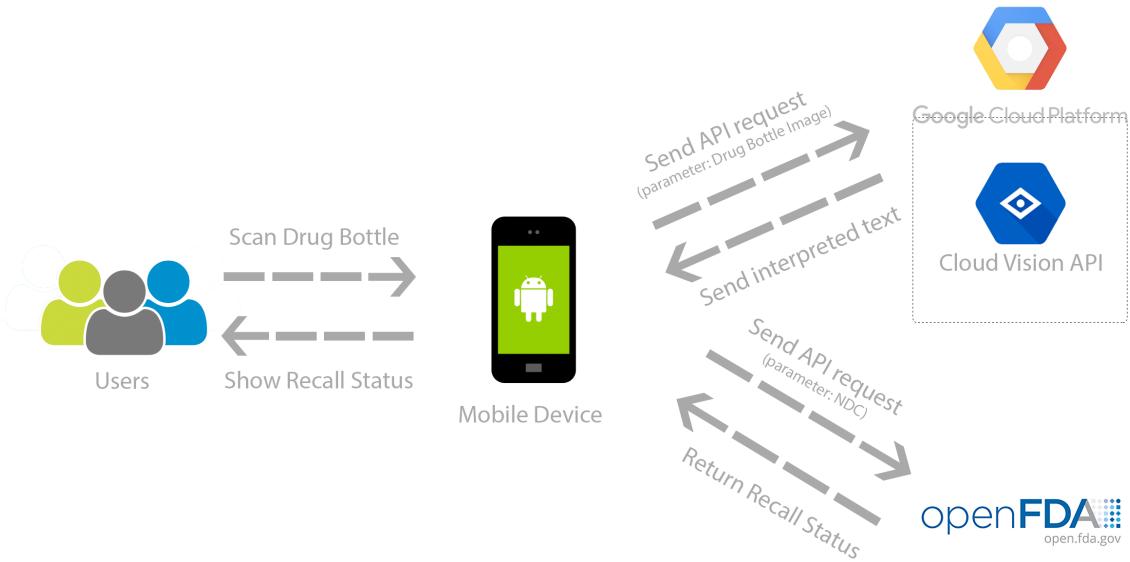


Medicine is rigorously tested for safety and effectiveness before becoming available to the consumer. In the U.S., the FDA makes sure this happens. Once on the market, the FDA, along with the makers of the drug, continue to monitor the medicine for any unforeseen problems. Should an issue develop, or the safety of a medication come into question, a recall may be initiated. We should stop taking the recalled medicine at once. Not everyone is educated about drug recalls in the market.

We are proposing a mobile application ***DrugWatch*** that will scan the medicine bottle and search the FDA database to determine if the scanned bottle is recalled drug or not.

ARCHITECTURE FLOW

Following diagram shows the Architectural Flow of our application:

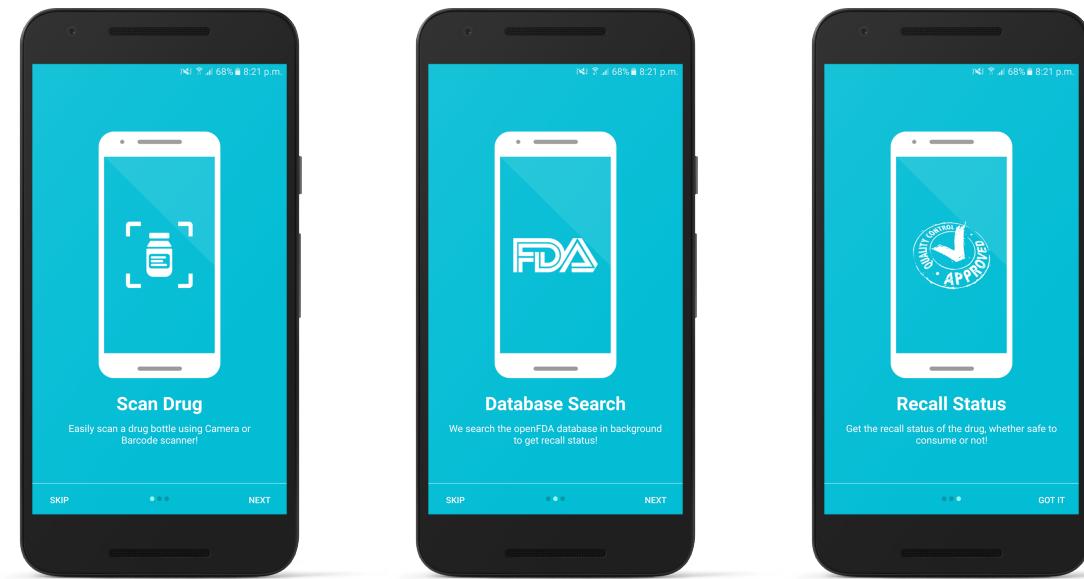


- User will first scan the drug using the available options from our android application.
- Android App will communicate with Google Cloud Platform using Cloud Vision API to send Image data.
- Cloud Vision API will extract the text from the scanned image and return that to android application.
- Android application will execute the algorithm to extract only National Drug Code (NDC) from the response.
- NDC will be used to query openFDA database using openFDA API.
- Depending on the response of openFDA API, android app will display if the drug is recalled or not with a popup message.
- If the drug is recalled, the popup will provide recall reason.

APPLICATION FLOW

The interaction between user and application as well as between application and cloud API's occurs in following order:

1. User launches the application and is provided with Usage Demo Screen.



2. Then the user is taken to main screen where he/she needs to select scanning option from the three available options.

- Camera
- Gallery
- Barcode

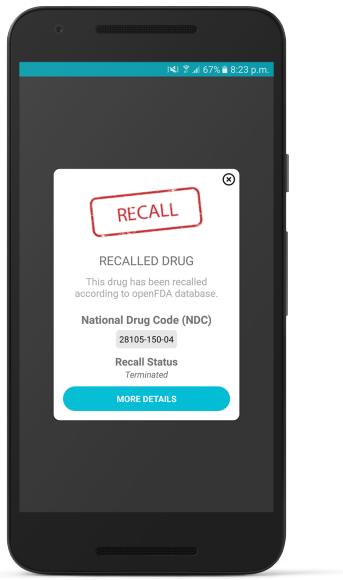


3. After select a scanning option user scans the Drug and can preview the scanned image. Image can also be rotated for better preview.



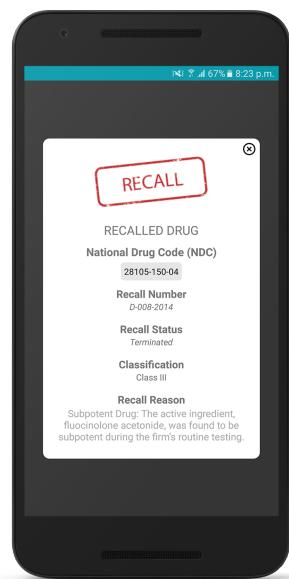
4. After the user submits a scanned image, application connects to Google Cloud Vision API and gets the text extracted from scanned image.
5. The response from Google Cloud Vision API is then parsed through an inbuilt custom algorithm to find any NDC in the text.
6. Depending on the availability of NDC in the text android application shows varied output popup's.
7. The output popup can be one from following:
 - Recalled
 - Approved
 - Sorry

- Once the NDC is detected it is queried with openFDA Database and API for recall information. If recall information is found, then Recalled Output is shown.

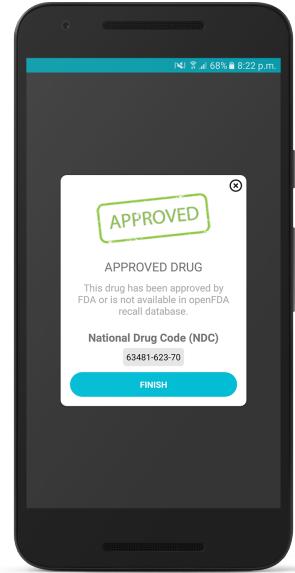


- When user selects more information a detailed screen with following information about recalled drug is shown to the user.

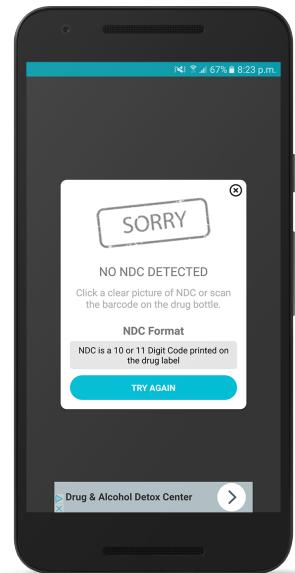
- Recall Number
- NDC
- Recall Status
- Recall Reason
- Classification



10. If recall information is not found in openFDA database, then Approved popup is displayed to the user.



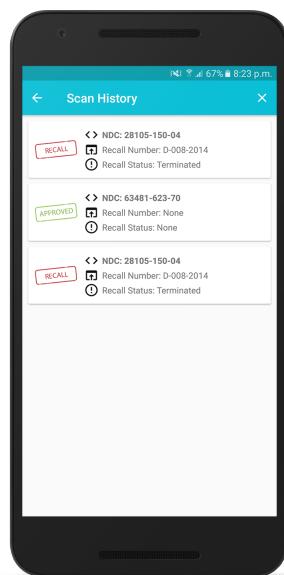
11. If no NDC was detected by the algorithm, then a Sorry message is displayed to user.



12. Next the user can select from various options provided inside the application.



13. If the user selects Scan History, then previous scan history by the user is shown in our application.



14. Usage Demo option will again show the Demo screen shown at the application start-up.
15. Share App option will enable the user to share our application with friends and family.

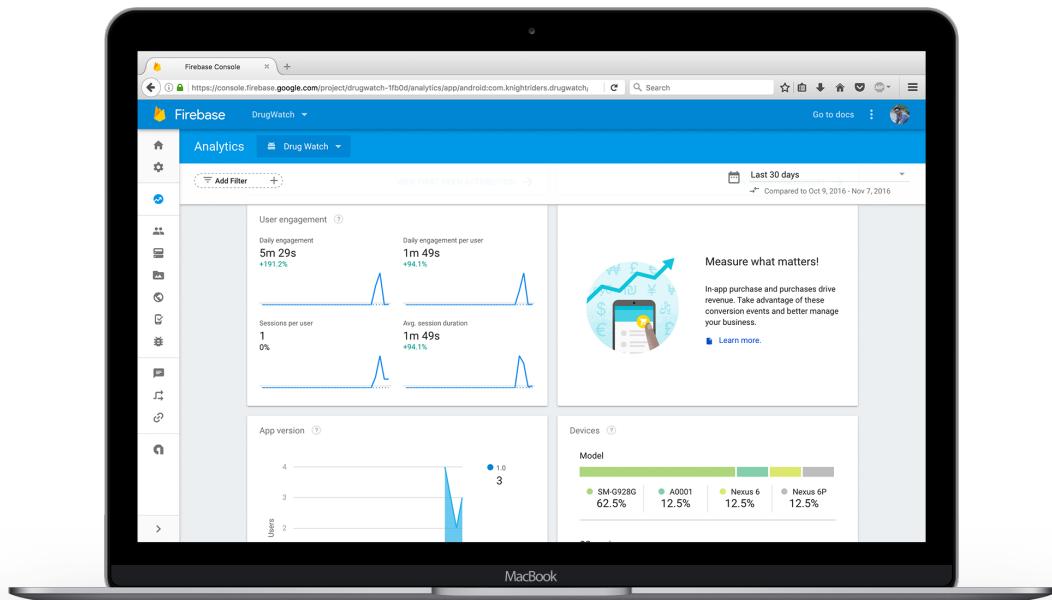
INTEGRATED DEVELOPER TOOLS

We have integrated various tools that will help the developer of the application manage and improve the application in future even if the user base increases by massive amount.

The integrated tools are:

- Firebase Analytics
- Firebase Remote Config
- AdMob

FIREBASE ANALYTICS



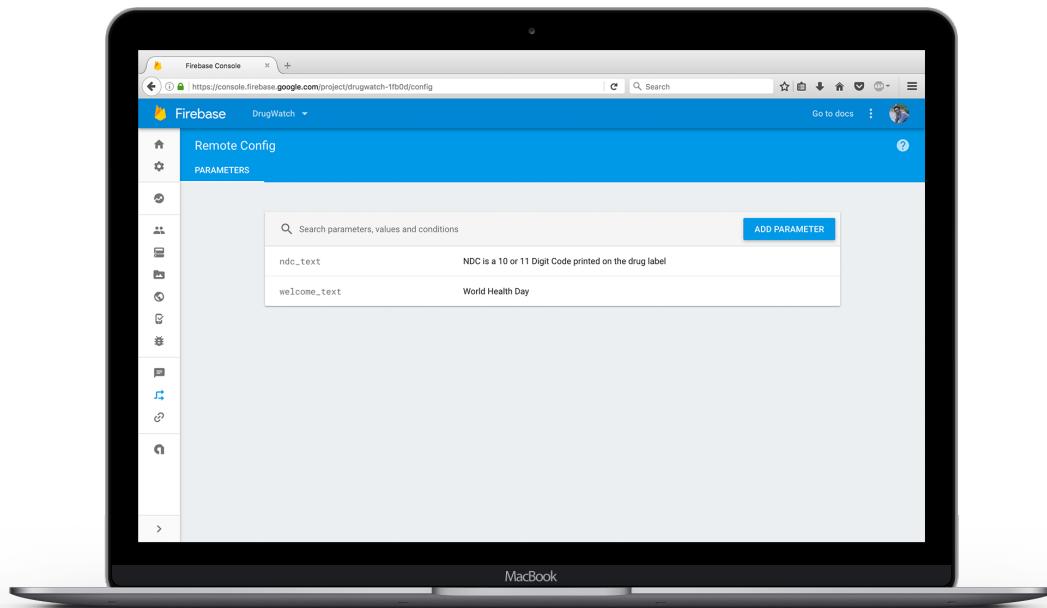
Firebase Analytics was integrated to keep a track on how the user interacts and which part of the application is keeping the users most engaging.

This information can be used to improve the user experience in future depending on various datasets provided by these analytics.

The data gathered from analytics include:

- Clicks
- User engagement
- Demographic details of users
- Device information
- Android version information, etc.

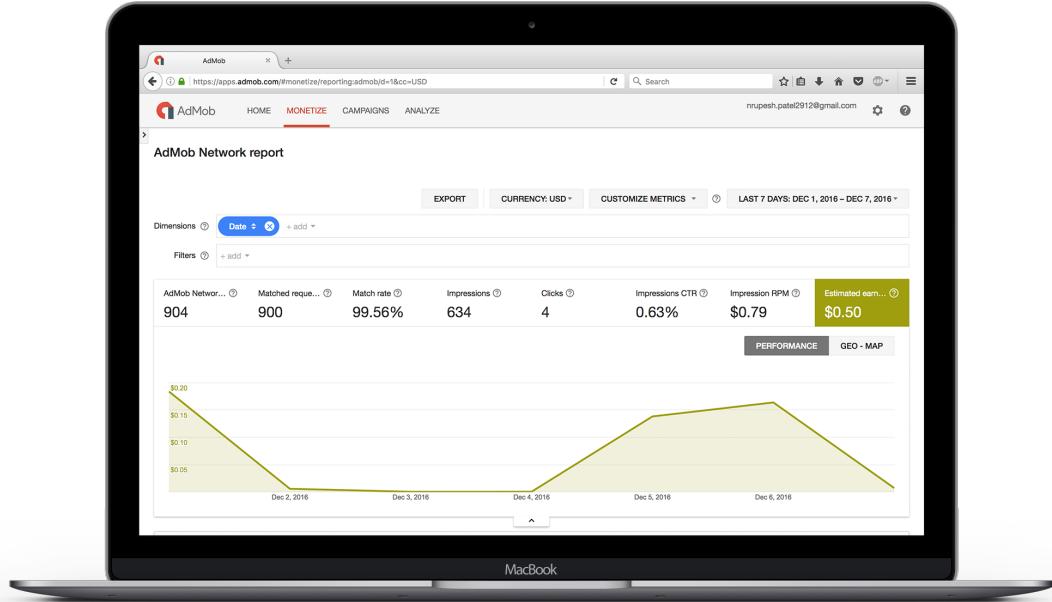
FIREBASE REMOTE CONFIG



Firebase Remote Config was integrated to enhance the user experience by not providing heavy updates for minor changes in the applications and instead pushing them directly into the application from cloud servers.

This can be used to customize the user experience during some special health related days by changing the theme of the application or displaying a custom welcome text for the user.

GOOGLE ADMOB



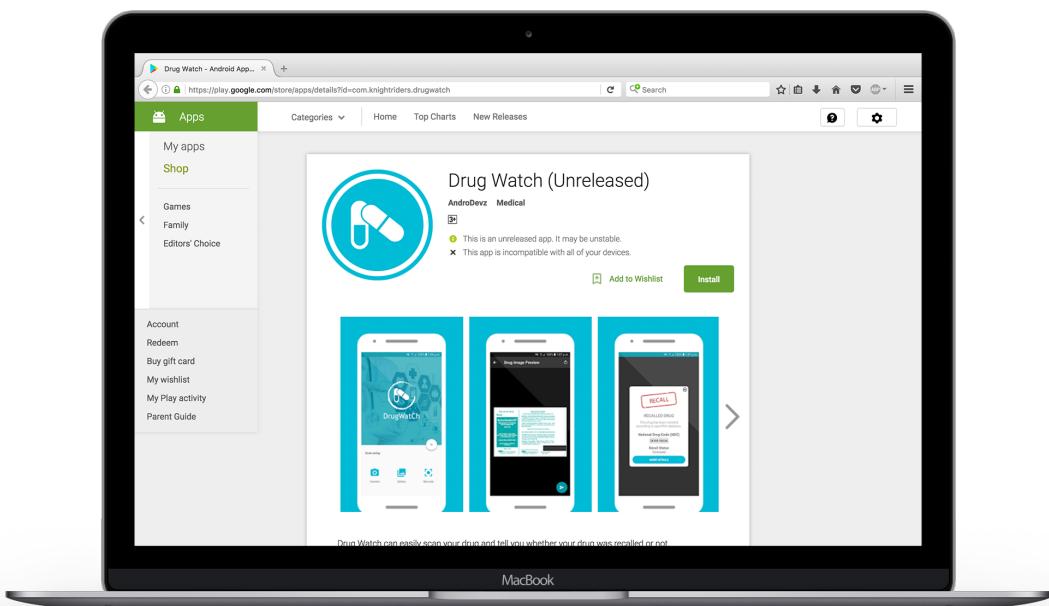
AdMob was integrated to generate revenue from the application without affecting the smooth user experience. Only a small banner ad is displayed after the recall information is displayed to the user.

The categories of the ads have been narrowed down to Health related domains in order to enhance the user experience.

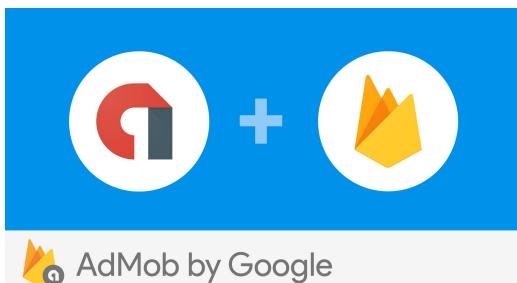
The Firebase Analytics and AdMob are linked so as to display ads to user which suits the pattern observed in Analytics data.

BETA TESTING

Our application is available for Beta Testing on the Google Play Store.



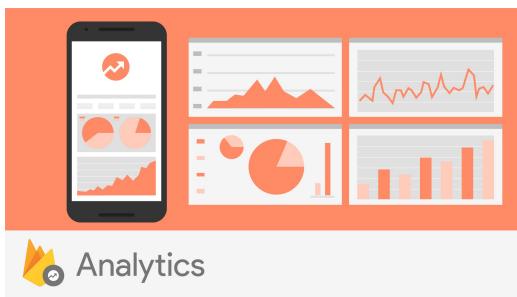
TECHNOLOGIES USED



AdMob by Google



Remote Config



Analytics

openFDA
open.fda.gov



Google Cloud Platform